cheaper and quicker

TIME-SHARING IN BIOMEDICAL RESEARCH

by T. ALLAN PRYOR and HOMER R. WARNER

The computer is fast becoming a useful tool for biomedical research. With this tool, as with others before it, potential users have the right to ask: how will it do my job better and how much will it cost? With these questions in mind the computer system to be described was developed to increase the ease with which certain kinds of biomedical research can be performed and minimize the cost and time for both the investigator and computer.

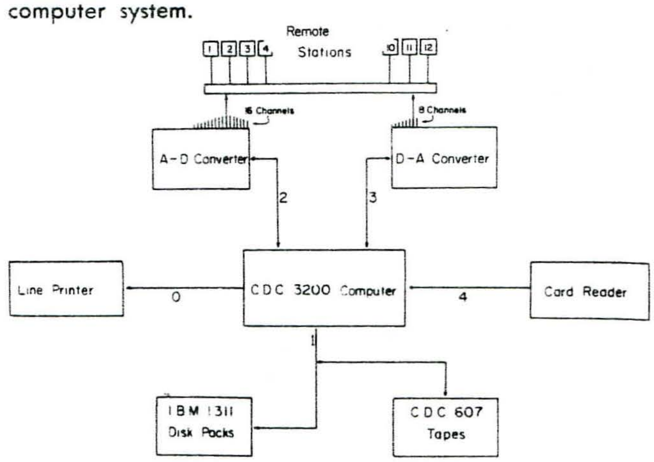
Some specific problems which arise in applying computers to biomedical research are: 1) the need for sampling of biological data over extended periods of time or sampling at very high rates for short intervals, 2) the separation of computer and experimental site, and 3) the need for continual rewriting of research programs as the form of the experiment is changed to accommodate a change in the mathematical model of the biological system. To solve these problems most efficiently a multiple remote station system has been developed. Each station can communicate with a central processor and is able to run experiments in conjunction with one to six other stations, one of which is the card reader and printer at the central processor. The system can process data from several experiments simultaneously as well as perform the more routine card and tape oriented data processing and computing. In most cases the monitoring of experiments takes relatively little computer time from the more conventional computer processing. A time-sharing monitor has been written to accomplish this. The hardware and software of the system, along with several applications, are described in this paper.

Fig. 1 is a diagram of the computer system at the Latter-day Saints Hospital. The system consists of a CDC 3200 computer with five I/O channels and 32,768 24-bit words of memory, and the following peripheral equipment: a high speed line printer connected to computer I/O channel 0, three tape drives and two IBM 1311 disc pack units on channel 1, an analog-to-digital converter connected to channel 2, a digital-to-analog converter on channel 3 and a 1200-card-per-minute card reader on channel 4. Channel assignment was based on job function in order to keep the channels as independent from each other as possible for maximal efficiency of the buffered I/O during simultaneous use of these peripheral devices. An analog computer is available for hybrid computations. It is connected to the central processor using one or more of the analog-to-digital, digital-to-analog channels.

conversion equipment

The analog-to-digital (A-to-D) converter is an 8-bit unit with a maximum conversion rate of 100,000 samples per second. At present there are 16 analog input channels and 16 digital input channels. The digital inputs are 12-bit binary words which are gated directly into the computer. Selection of the analog or digital channel to be used for a particular program is under program control. Various modes of operation are available to the programmer in-

Fig. 1 Block diagram of Latter-day Saints Hospital real-time computer system.



Mr. Pryor received his BS from Long Beach State College and his MS from the University of Utah. He is presently systems director of the digital computer facility at the Latter-day Saints Hospital and assistant research professor of the Department of Biophysics and Bioengineering at the Univ. of Utah.

Supported by a grant from the National Institutes of Health, United States Public Health Service, No. 5 PO7 FR-00012.

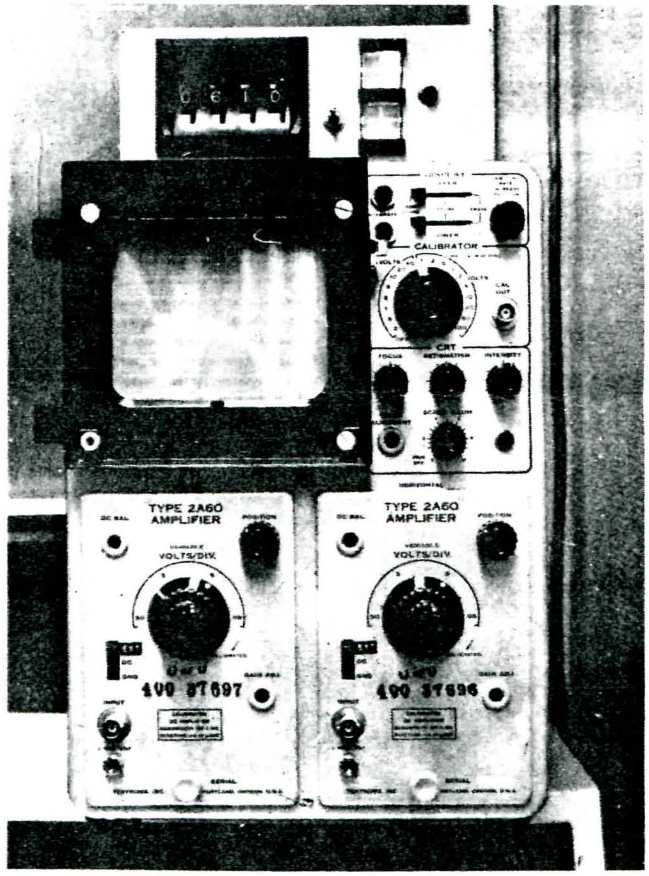
cluding interrupt modes which allow the A-to-D converter to send to the computer an external interrupt when data is ready to be transferred to the computer. Sampling rates of the A-to-D converter can be controlled both internally by the computer or externally through a pulse or sine wave fed directly to the A-to-D converter. In this external sync mode, data will be transferred to the computer only when the A-to-D converter is ready and some other external sync pulse has been supplied. There is also a hold-off feature which restricts sampling and transferring of data from the converter into the computer until an external hold-off pulse has been received by the A-to-D converter. After this pulse has been received, the converter will transfer data in either the internal or external sync mode. Connected with each digital input channel is a four-digit thumbwheel octal switch through which an operator is able to transfer digital information to the computer.

The digital-to-analog (D-to-A) converter has a maximum rate of 100,000 conversions per second and will convert the low-order 8 bits of a 12-bit digital word to an analog voltage varying between 0 and +10 volts. The present system consists of eight analog channels connected to the D-to-A converter. The first two of these channels are hard-wired directly into six Tektronix memory scopes located in the different laboratories. All information displayed on these scopes comes via these two analog channels. The erasing and blanking of any scope is controlled by the program through the use of three banks of relays available on the D-to-A converter. Thus, although data is transferred to all the memory scopes, it will, by use of the relays to control blanking, be visible only on the particular scope whose station has requested the information. The relays not only control the erasing and blanking of the scopes but also control the internal and external sync modes of both converters.

The investigator communicates with the computer through remote stations. Each of these (see Fig. 2) consists of a computer interrupt push button, a digital input dial, a memory scope, and eight small indicator lights. These lights indicate the status of the computer, i.e. which areas of core memory contain active programs and the priority of programs in core or waiting on disc to be run. By pressing the interrupt button with an appropriate code word in the digital input dial, the investigator may request his program be brought into memory. Once in memory the program normally writes on the scope in the laboratory of the investigator the sequence he is to follow. Subsequent interrupts from his station then allow him to communicate with his program and to start and stop the flow of information through the converters.

At locations remote from the hospital the data is transferred to the central processing area via telephone lines.

Fig. 2 A remote station consisting of memory oscilloscope, octal switch, indicator lights and interrupt button.



Here the investigator, instead of feeding his data directly into the A-to-D converter, feeds his analog data into the voltage controlled oscillators and a frequency multiplexer. The FM carriers are then transmitted via a private telephone line to the central processing location. Here the FM signals are demodulated and fed into the A-to-D converter for input into the computer. The results of the experiment are in turn relayed back to him using a similar system over another telephone line.

Dr. Warner received his MD from the University of Utah and Ph.D. in physiology from the Univ. of Minnesota. He is presently director of the Cardiovascular Laboratory at the Latter-day Saints Hospital and professor and chairman of the Department of Biophysics and Bioengineering at the Univ. of Utah.

basic programs

Because an investigator may need repeated access to the computer during a long experiment, a monitor program has been developed to control the use of the computer and allow simultaneous experimentation from more than one remote station. Programs written to communicate with the remote stations are written in assembly language. These programs may, however, act as subroutines for a FORTRAN program. Where extensive computation is required on the real-time data, the computational portion of the program is written in FORTRAN and the communication with the A-to-D, D-to-A and remote stations in assembly language.

All user programs are stored on one of the disc packs. When a program is requested, the monitor brings it into core for execution. Through a program priority system, it is possible for one program to be interrupted during execution and another program of higher priority to be initiated. At this time the program of lower priority is stored in a temporary area on disc to be completed later.

A central print driver has been written to prevent the printed results of one program from being intermixed with another on the line printer. Output from every program, as well as compiled listings, is placed on the disc. At the end of execution of a given program the monitor initiates the disc-to-print program, which then prints in an interrupt mode the entire results of that program. After completion of one print job, the system looks to see if other programs have been completed and require some listing. If so, it initiates the listing of that program. Only FORTRAN programs are allowed to have printed output. The output of the assembly language programs will be to the scopes.

Access to the disc is via a central disc driver. This is to eliminate the moving of the disc arms by many programs causing a loss of data.

Programs written in FORTRAN are compiled into the lower 16K of core. Since they require so much core even for a relatively short program, a priority scheme was developed to use this portion of core most effectively. Priority three was given to FORTRAN-compiled programs which require more than five minutes to run while priority two programs run in less than five minutes. A FORTRAN-compiled program is priority one if it is linked with upper core to one of the assembly language programs. These require highest priority since they use data in real-time from some experiment in progress. This option requires that the program be accessible to the investigator within two seconds in order to insure no loss of data. Although these programs are of highest priority they are swapped on the disc during times of no activity from the station using the high priority program.

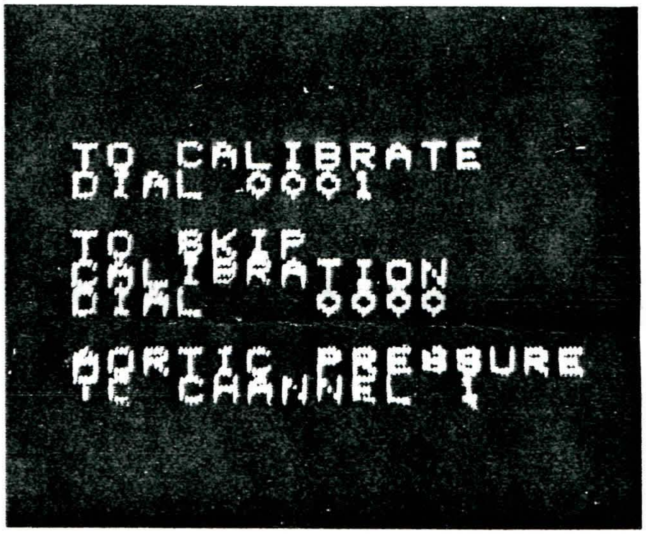
Above this first 16K area the next 12K is set aside for assembly language programs which handle real-time data from the laboratories. These programs are designated real-time programs. These programs are relocatable in this area and upon execution are brought into some part of this 12K area. They remain in core, however, until the experiment is finished or the investigator sends an appropriate code to the computer. All programs in this area are called from the remote stations by depressing the interrupt button after dialing 7XXX in the octal switch, where XXX is the program number. The monitor interprets this as the beginning of program XXX, reads it from the disc into memory and transfers control to the program.

Again, the main function of these upper core programs is to communicate with the experimenter and do the actual transferring of data to and from the computer. For example, these programs may write a control message on a memory scope (Fig. 3) to tell the experimenter to connect his input to certain analog channels or to specify a program option desired. Upon completing these functions the experimenter then presses the interrupt button telling the computer that he has performed the function and is ready to proceed with his experiment. This two-way communication proceeds with the operator controlling the flow of information from the experiment to the computer and the computer asking for data and options in the proper sequence and finally displaying results on the oscilloscope as alphanumeric characters and or graphs. A more detailed description of several of the programs will be given further on in the paper. The monitor can control A-to-D and D-to-A operations for six such programs in memory at any one time.

The upper 4K words of memory contains the monitor and several special-purpose subroutines which are accessed by all of the real-time programs. One such program is the subroutine to write alphanumeric information on a scope.

Multiplexing between the programs in memory may take place as a result of an interrupt from a remote station or from the real-time clock within the computer. The clock interrupts are normally used to control sampling rates. Each program specifies to the monitor the desired interval between successive samples of its analog input-output. This interval is added to the reading of a 10KC internal clock and placed in an interrupt mask table to switch control to that program when the next sample is due. If in any program there is a time delay for operator intervention or for the next sample to be ready, control is returned to the monitor which then tests whether there are other programs to be serviced.

Fig. 3 A message written on a memory oscilloscope giving procedural instructions to the user.



transfer interrupts

Since each of the real-time programs generates so many interrupts requesting that control be transferred to its program, it has been found that there need be no scheduled cycling through the programs by the monitor. The only program not generating interrupts is the FORTRAN program in the lower 16K; but since only one such program is allowed to be executed at any time there is no need to interrupt it periodically to look for other similar programs in core since they do not exist. When a FORTRAN program has been interrupted by some real-time program it takes from 0.1 to 30 milliseconds for the computer to complete the function required by the real-time program and return to the FORTRAN program. There is a small probability of a real-time program being interrupted at a time in the program when it is no longer generating interrupts by an interrupt which initiated a FORTRAN program, thus causing a long delay before returning to complete the real-time program. It has been found through experience, however, that the probability of this happening is negligible. For this reason, cycling periodically through the various programs to determine if they require servicing would only reduce computer efficiency.

FORTRAN-compiled programs are debugged in the normal manner using the compiler diagnostics and a sufficient number of print-outs in the program to inform the programmer of his errors. Debugging the real-time assembly language programs is accomplished on-line through the use of a special debug program. This program uses a memory scope and the on-line typewriter to communicate with the programmer and his program. It permits the programmer to execute a single instruction, read and write in core and execute specified portions of his program without actually stopping the machine or interfering with other experiments in progress. All options available at the computer console are available to the programmer at the typewriter. At each breakpoint in the program all registers are displayed to the programmer and some option requested. This on-line debugging is considered a necessity if the system is to be a true time-sharing system.

A few programs are not compatible with this time-sharing mode of multiple simultaneous program servicing in that they use an extremely high analog data sampling rate, or special A-to-D synchronizing schemes. An example of such a program will be described later. These programs occupy the computer for only a few seconds and can be scheduled at specific times during the day. To illustrate the use of this system, some programs will be described here which use various features of the system and demonstrate the flexibility which is available at the present time.

examples of users' programs

The first example of a program involves solution of a mathematical model developed to describe the transfer function of pressure receptors in the wall of an artery. The input to these receptors is the pressure in the artery and the output is in the form of nerve spikes or action potentials on the nerve going from the organ to the brain.[1] To test this model, arterial pressure and frequency of nerve firing are sampled from an anesthetized animal. This data is then compared with the frequency of nerve firing generated by the mathematical model. To obtain the data two programs have been written. The first determines the number of nerve fibers which are being detected by the pick-up electrodes. This is accomplished by sampling 1000 action potentials and then returning to the investigator an oscilloscope plot of a histogram of the spike amplitudes. It has been shown that a single fiber fires at essentially a constant amplitude. Amplitudes will vary between fibers due to differences in electrode contact with the fiber. However, because of noise in the recording system, the amplitude histogram of a single fiber has the form of a normal distribution. Upon receiving a histogram, the investigator makes a judgment as to whether there is one or more fibers being recorded. If there are more than one, he continues to dissect the nerve until he obtains a histogram which is unimodal. Here the computer serves as a valuable tool for real-time work. The output of the nerve could easily be stored on magnetic tape and processed at a later date but if due to some error the correct data was not recorded, the experimenter has lost both time and information. With the computer available to him during the experiment, he advances to each new step with the results of the last step already verified.

To obtain the histogram described above, the output of the recording electrodes is fed into a digital logic system which is set to trigger on a given threshold. When a nerve action potential crosses this threshold, it triggers a 500μsec-duration pulse. The computer meanwhile has requested an input of 25 samples from the A-to-D with the converter set in the external sync mode. A sine wave is "anded" with the output of the digital logic system and is fed into the A-to-D converter only during the 500μsec when the nerve spike is above the threshold. The 55KC sine wave is allowed to synchronize the input of analog data into the computer during this time. The computer is programmed to interrupt when the 25-word input buffer is full. This will occur in slightly less than the 500μsec. The maximum amplitude of the action potential will then be determined from these samples and stored for later display in the form of an amplitude histogram. Since the input/output of the computer is buffered, the computer is able to service only those other programs which do not use the A-to-D converter while waiting for a nerve firing. After 1000 action potentials have been so processed, the program calculates an amplitude histogram and displays this via the D-to-A converter as a plot on a memory scope at the investigators station.

Once the investigator is satisfied that he is recording from a single nerve fiber, he calls a second program. This program samples the input to this organ, the arterial pressure, at a rate of 100 samples per second. As it reads in the pressure wave it determines the beginning and ending of each heart cycle. Corresponding points on each succeeding pulse are added and the resulting average waveform is then stored in memory. At the same time the voltage from the nerve is fed through another digital logic system which causes an external interrupt to the computer each time the nerve potential crosses a set threshold. Upon receiving these external interrupts the program reads the real-time clock to determine the frequency of firing at that instant of the heart cycle. Thus, for each averaged point on the pressure wave, an averaged frequency of firing at that point in the cycle is also calculated. Simultaneous with these calculations the averaged pressure wave and frequency of firing are displayed on an oscilloscope at the investigator's station 50 times every second. This, then, enables the investigator to follow the build-up of these averages. When displayed averages become stable the investigator generates a manual interrupt via his station which causes the computer to write the averaged data on a digital tape. At the end of an experiment the investigator may call a FORTRAN program which tests a mathematical model of the organ. This program predicts the time-course of frequency of nerve firing from the time-course of arterial pressure. This predicted frequency of nerve firing is then compared with the actual frequency of nerve firing obtained from the animal as a measure of the validity of the model.

The first two programs described here may either be run on-line real-time, that is, feeding the data from the animal directly into the computer, or the data may be run from analog tape. Since the histogram program uses a special synchronizing scheme with the A-to-D, it must be run at a time when no other real-time programs are being processed. This does not mean that another program cannot be run during the complete experiment, but only during those few seconds that the program is sampling data.

analog processing

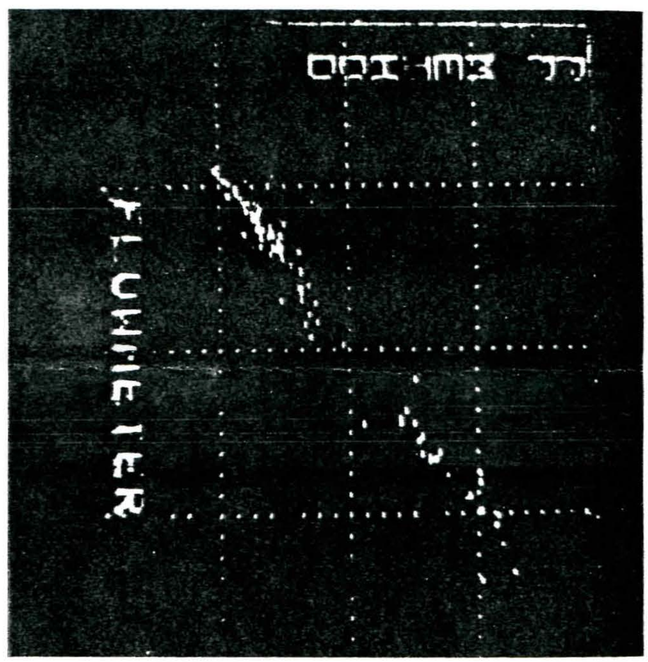
In many instances in the formulating of a mathematical model for some biological system, it is more convenient to program the mathematical model on an analog computer. The investigator tries to match by empirical adjustment of equation parameters the output of the model on the analog computer with actual data which has been recorded from an animal. This process becomes quite time consuming since the investigator must test the model against data obtained under a variety of physiologic states before he can be confident that it is an adequate description of the system. From the raw data, such as blood flow and pressure, many variables must be calculated, such as heart rate, cardiac output, stroke volume, resistance and mean arterial pressure. The time-sharing system allows the digital computer to do much of the work that would normally be done by the investigator at small expense to the computer. A program to do this has been written in two sections. The first part inputs from an analog tape or

[1] Christensen, B. N., Pryor, T. A., and Warner, H. R. "A Technique for the Quantitative Study of Carotid Sinus Behavior." *The Physiologist* 8:134, 1965.

directly from the animal the raw data and calculates the derived variables for a time interval specified by the investigator. It then logs this data on a digital tape for replay later. When the investigator has programmed his model on the analog computer he is then able to call the second portion of the program which reads these variables from the digital tape. With each sweep of the oscilloscope a clamp pulse is generated which initiates the solution in the analog computer and acts as an interrupt to the digital computer. The digital computer outputs these variables through the D-to-A, some acting as forcing functions for the analog computer model and others being displayed on an oscilloscope for comparison with the corresponding variables being predicted by the analog computer model. The output from the digital computer may be at any multiple of real time requested via the digital switch. It would be difficult to justify this mode of operation were it not for time-sharing and multiple-station operation because of the inefficient use of the 3200 by this single program. However, under the multiple-station processing mode, this becomes an effective use of machine time as well as the investigator's own time in developing his mathematical model.

Continuous monitoring of various parameters from an experimental animal is another task readily accomplished by the multiple-station, time-sharing approach. An example of a program to do this is the pressure pulse stroke volume program which provides a means for estimating the amount of blood ejected by the heart on each beat from the contour of the pressure wave recorded in the aorta.[2]

Fig. 4 Comparison of stroke volume determination by pressure pulse method and flowmeter method.



Within ten milliseconds after the end of each heart cycle, an analog voltage is generated to represent each of five variables calculated from the pressure waveform and the input of data can continue indefinitely. The method is based on the fact that the aorta is distensible and acts as a capacitor. Part of the blood that is ejected by the heart

[2] Warner, H. R., Swan, H. J. C., Connolly, D. C., Tompkins, R. G., and Wood, E. H. "Quantitation of Beat-to-Beat Changes in Stroke Volume from Aortic Pulse Contours in Man." *Journal of Applied Physiology* 5:495, 1953.

in each cycle is stored in this distensible tube during the ejection phase; then, during diastole, when the heart is not contracting the stored blood runs out of the arteries as the pressure wave decays. The program consists primarily of a scheme for recognizing the onset of systole and the end of systole or dicrotic notch. The difference in pressure as a function of distance down the aorta at these two points in time is estimated and this pressure difference is related to the difference in volume of the aorta at these same two points in time. The ratio of flow out of the system during systole to flow out of the system during diastole is considered to be proportional to the ratio of the time integrals of pressure over these two periods in time. Then by estimating a single constant which relates stroke volume as measured by the flowmeter to stroke volume as estimated by the pressure pulse method with the dog in a resting state, subsequent beat-by-beat estimates of stroke volume can be made under a variety of physiologic states.

Fig. 4 shows a cross-plot of beat-by-beat estimates of cardiac output made by the pressure pulse method against the measured stroke volume obtained by integrating the output of the flowmeter curve. The correlation between the two methods varies from .92 to .98 under a wide variety of physiologic states including exercise and the infusion of drugs which raise and lower blood pressure and raise and lower heart rate. This continuous monitoring in humans of five important variables in the circulation from a single pressure input is done by using only a small amount of the computer time since other programs are being processed at the same time but still insures accurate pattern recognition of the pressure and flow curves..

At one of the remote stations a typewriter and memory oscilloscope are available for on-line use with the computer. A standard use of the typewriter is entering parameters in mathematical models of various biological systems. The time-course of any variable may be displayed as a function of time on the memory scope and compared with a previous solution. The FORTRAN portion of the model is swapped in and out of memory by a small control program which resides in upper core and controls the communication with the typewriter and the oscilloscope. Thus, the FORTRAN area of core memory is available for other programs except when a new solution of the model is requested. Variables and parameters in the FORTRAN program are referenced through indirect addressing in the control program. The program allows the operator to easily explore the effects of variations in each of the parameters on model performance and is an effective tool for designing experiments and interpreting results.

system evolution

Although this system has been in operation for about 18 months in this laboratory, it has undergone a gradual evolution over this period and will continue to evolve to meet the changing needs of the research community. The programming staff of the laboratory consists of two systems programmers. The rest of the programming is all done by individual researchers and graduate students. The system has recently been put into use on the 3200 system at the Mayo Clinic and will be made available to all 3200 users through Control Data; the company has accepted responsibility for system maintenance. The system is designed to fill a specific need for those engaged in biomedical data of the type generated in a physiology laboratory and is not meant to do all things for all people. Primarily because of this, the system runs efficiently and was implemented by a few people in a short time. An effort is now being made to make the system more readily available to a larger portion of the medical research community in Salt Lake City through expansion of the remote telephone terminal concept. ■